

The Data Assimilation Research Testbed (DART)

Informal Workshop on Emerging Practices for
Component Interfaces in Earth System Models

14 May, 2003

What is DART?

1. Allows combinations of assimilation algorithms, models, and observation sets
2. Diagnostic tools
3. Data Assimilation R&D for NCAR and external partners
NOT for operational use or support

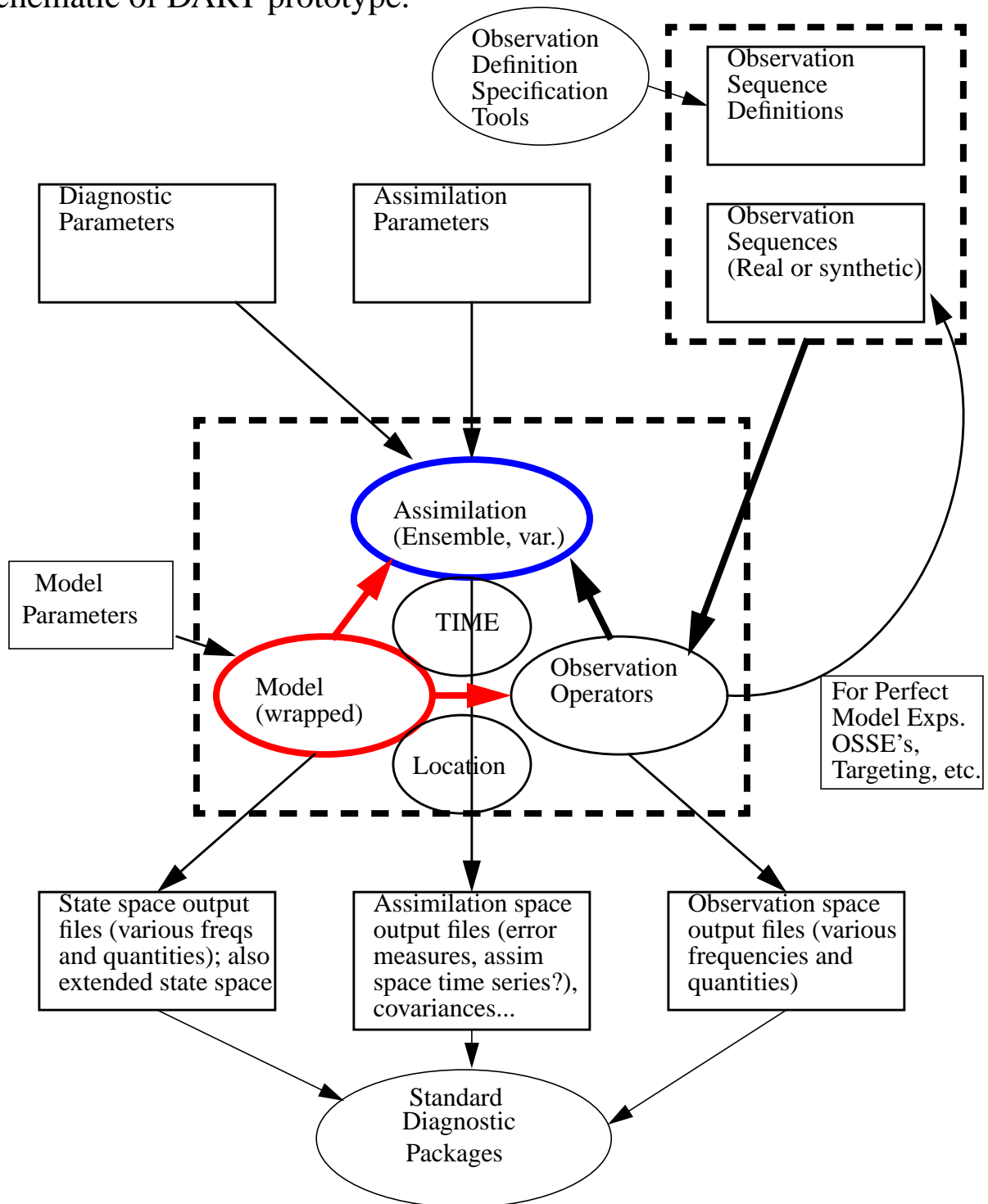
Status of DART

1. Basic framework implemented
2. Currently using GFDL FMS infrastructure
3. Switch to ESMF infrastructure when available
4. Primarily implementing ensemble (Kalman) filters
5. Variational for low-order models only
6. Plans MAY include a variational (4D-Var) capability

DART compliant models

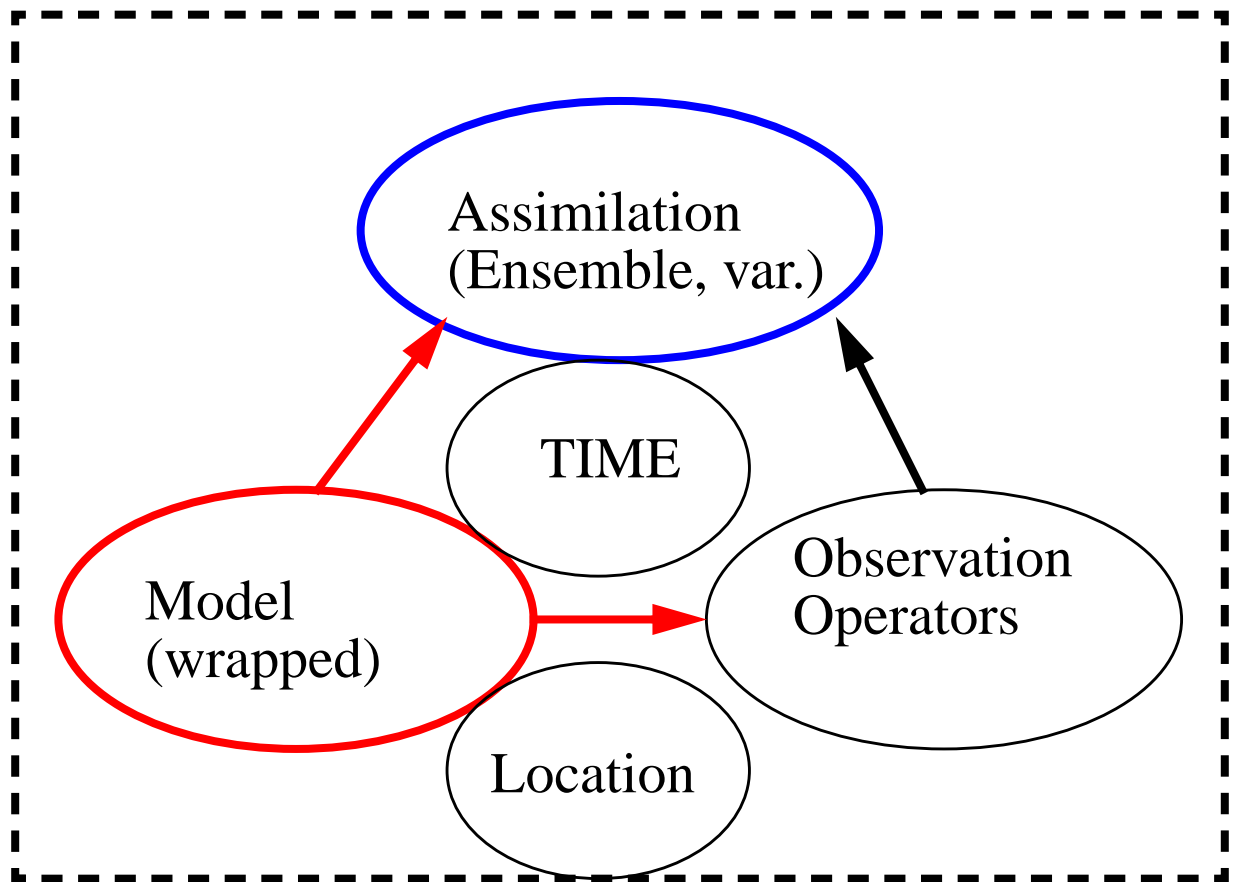
1. GFDL FMS B-grid GCM incorporated and in use
2. NCAR CAM 2.0.1 incorporated and being tested
3. Many low-order models available
4. WRF model in process of being incorporated
5. NCEP MRF tested quasi-operationally, partial implementation

Schematic of DART prototype.

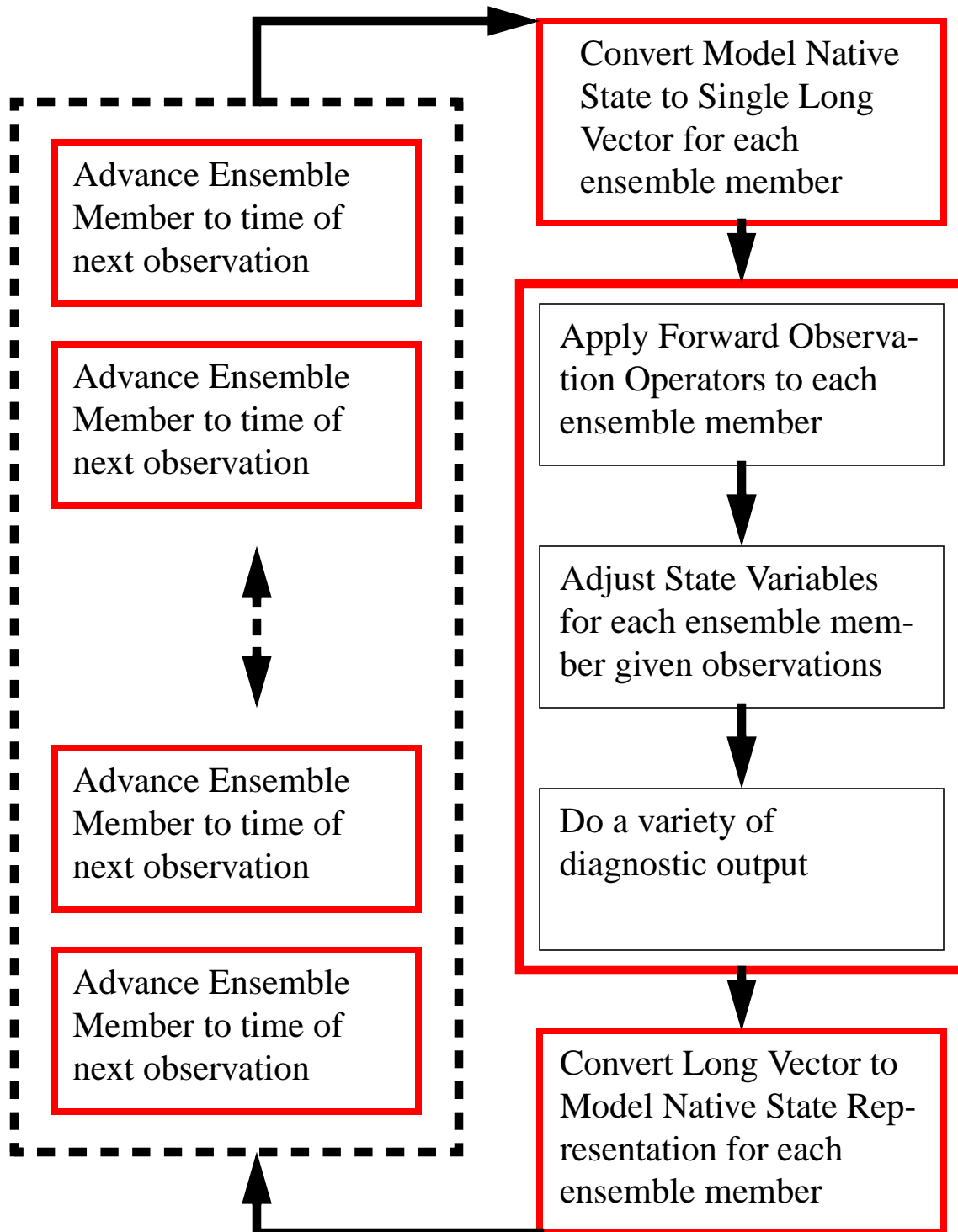


Concentrate on ‘interfaces’ to **model**

Try to focus on fundamental interfaces



Control Flow for Ensemble Filter interaction with model



Red boxes indicate possibly independent executables

Dashed black box indicates parallel execution permitted

Note for CS types:

Geophysical model is finite state machine here

What things do we need from models?

Two basic tasks:

A. Advance model in time

B. Access and modify model state

Requirements for models:

- 1). Clear and succinct definition of model state vector x

$$x(t_2) = M[x(t_1), t_1, t_2]$$

Note: x may contain multiple time levels

- 2). Handle to model state, (data and/or file structure)
- 3). Ability to advance model given state representation, and selected values of t_1 and t_2

Required at 'shell' level

Nice as a Fortran callable subroutine but not requirement

- 4). Ability to restart EXACTLY from state

$$x(t_3) = M[x(t_1), t_1, t_3] = M[M[x(t_1), t_1, t_2], t_2, t_3]$$

(given that single run has a time-step boundary at t_2)

Requirements for models (cont.)

- 5). Clear and succinct definition of static model ‘parameters’

Really have $x(t_2) = M[x(t_1), t_1, t_2; y]$

Includes boundary conditions and parameter settings
Everything should be modifiable at ‘run-time’

- 6). ‘Run-time’ access to info about time-stepping capabilities

Bonus: ‘Run-time’ control of time-stepping if applicable

- 7). ‘Run-time’ access to model state metadata in standard format

- 8). ‘Run-time’ access to parameter metadata in standard format

Requirements for models (cont.)

9). Flexible state representation where applicable

(spectral, linear grid, non-aliasing grid???)

10). Flexible ability to select additional non-state variables for output at restart times with associated metadata

Precipitation as an example

11). Diagnostic output: Fortran callable methods to:

A. Output metadata for state vector in standard format

B. Output state vector in corresponding format

12). Adjoint or compliance with standard adjoint generating tools for variational capability (are we kidding???)

Conclusions:

1. Most requirements are in basic design
2. Treat model as a piece of software someone external might use
3. Standardizing formats for interfaces would be additional help but is not essential